Compressive Video Sensing via Dictionary Learning and Forward Prediction

Nasser Eslahi, *Student Member, IEEE,* Ali Aghagolzadeh, *Senior Member, IEEE*

and Seyed Mehdi Hosseini Andargoli, *Member, IEEE*

arXiv:1508.07640v1 [cs.MM] 30 Aug 2015

Abstract

In this paper, we propose a new framework for compressive video sensing (CVS) that exploits the inherent spatial and temporal redundancies of a video sequence, effectively. The proposed method splits the video sequence into the key and non-key frames followed by dividing each frame into the small non-overlapping blocks of equal size. At the decoder side, the key frames are reconstructed using adaptively learned sparsifying (ALS) basis via ℓ_0 minimization, in order to exploit the spatial redundancy. Also, three well-known dictionary learning algorithms are investigated in our method. For the recovery of non-key frames, a prediction of current frame is initialized, by using the previous reconstructed frame, in order to exploit the temporal redundancy. The prediction is employed into a proper optimization problem to recover the current non-key frame. To compare our experimental results with the results of some other methods, we employ peak signal to noise ratio (PSNR) and structural similarity (SSIM) index as the quality assessor. The numerical results show the adequacy of our proposed method in CVS.

Index Terms

Compressive Video Sensing, Sparse Recovery, Split Bregman Iteration, Dictionary Learning.

I. Introduction

Conventional approaches to image/video compression usually have high computational complexity in encoding but they remain simple in the decoding process part. Also, distributed video coding (DVC) [1] is a low complexity technique which refers to a special video coding paradigm that encodes frames

of video sequence independently (the encoder can be very simple) and decodes them jointly at the expense of a more complex decoder (due to exploitation of the temporal redundancies by the decoder side). In both conventional video coding and DVC, data collection and compression tasks are performed disjointedly, with a high cost mechanism that wastes most valuable acquired data, because of limitation of allocated power and available memory. Due to great efforts by Candès *et al.* [2], [3] and Donoho [4], compressive sensing (also called compressed sensing or compressive sampling) suggests a new framework for simultaneous sampling and compression of signals at a rate significantly below the Nyquist rate. It also permits that under certain conditions, the original signal can be reconstructed properly from a small set of measurements via solving a convex optimization problem or iterative greedy recovery algorithms.

Recently, the idea of compressive sensing (CS) for imaging (single pixel camera [5, 6]) has been extended to the conventional predictive/distributed video coding, to develop highly desirable compressive video sensing (CVS)/distributed compressive video sensing (DCVS). CVS employs both data acquiring (video sensing) and compression into a unified task which emerges a new procedure to directly acquiring compressed video data via random projection for each individual frame, in a low-complexity encoder.

A. Related Works

Several CVS recovery methods have already been proposed, i.e., Wakin *et al.* [7] proposed an intuitive (motion JPEG motivated) approach which extends compressive image sensing to video applications by considering each frame of the video sequence independently, and recovers each frame using the 2D discrete cosine transform (2D DCT) or a 2D discrete wavelet transform (2D DWT), individually. However, this simple extension fails to address the temporal redundancy in video; nevertheless compressed image sensing techniques explore the spatial redundancy within an image. To enhance the signal sparsity in both spatial and temporal domains and achieve higher sampling efficiency, several frames can be jointly considered as a signal and recovered under a 3D transform (e.g., 3D DWT) [7, 8]. Results show that a 3D video reconstruction (joint frames) using a 3D DWT is better than the 2D frame-by-frame reconstruction using a 2D DWT, but it incurs high computation cost and large memory requirement.

Park and Wakin [9] proposed a multi-scale recovery approach, which several CS measurements are taken independently for each frame, and also the motion estimation is applied at the decoding step. The recovered video at coarse scales (low spatial resolution) is used to estimate motion which is then used to enhance the recovery at finer scales (high spatial resolutions). However, its coarse-to-fine reconstruction computational cost is high. The same approach based on using two-steps to iteratively update the estimates for the images in the video and the inter frame motion, was proposed in [10]. Also, Cossalter *et al.* [11]

considered the motion estimation in their proposed joint compressive video coding and analysis scheme.

Stanković *et al*. [12] and Prades-Nobet *et al*. [13] proposed a block-based selective video sampling scheme where firstly divides frames of the video sequence into reference frames (or key frames) and non-reference frames (or non-key frames); then each frame is divided into the small non-overlapping blocks of equal size. The reference frames are sampled fully using conventional video compression techniques like MPEG/H.264 (intra encoding). In [12], the reference frames are used to predict sparsity of the successive non-reference frames and CS is applied only within the blocks that are predicted as sparse ones, whereas the remaining blocks are sampled fully. In [13], non-key frames are projected and recovered using CS techniques, with an adaptive redundant dictionary built by picking a set of local (spatially neighboring) blocks which are extracted from its co-located blocks in the previous decoded key frame, to form its basis without training. A similar approach was proposed in [14]. Nevertheless, such local dictionary-based basis may not work very well for blocks with large motion or when the entire scene undergoes motion translation. In addition such schemes highly rely on the qualities of the neighboring reconstructed key frames and the performance may be degraded due to poorly reconstructed neighboring key frames [15].

The work in [16] relies on small inter-frame differences together with a spatial 2D DWT to produce a sparse representation of the underlying video. Similarly, Zheng and Jakobs [17] proposed a video compressive sensing method that relies on the sparsity properties of video in the spatial domain, where key (reference) frames are fully sampled and measurements are applied to the difference between each pair of successive frames. The video signal is reconstructed by first reconstructing the frame differences using ℓ_1 minimization algorithm, and then adding them sequentially to the reference frame. However, they do not sufficiently remove the vast amount of temporal redundancy, e.g., when exist a large inter-frame difference and fast motion between adjacent frames.

A multi-hypothesis (MH) prediction approach for CVS was proposed in [18], where MH predictions of the current frame are generated from one or more previously reconstructed reference frames, and then combined to yield a composite prediction superior to any of the constituent single-hypothesis predictions.

Another dictionary based approach is presented in [19], where the dictionary is learned from a set of blocks globally extracted from the previous reconstructed neighboring frames together with the side information generated from them. This work has been extended to assign dynamic measurement rate allocation (for different local regions) by incorporating a feedback channel [15]. Haixao *et al*. [20] proposed a redundant dictionary generation scheme for compressed video sensing, which follows the sparse representation approach of [13]-[14]. The authors extended their work to present a maximum-likelihood dictionary learning based reconstruction algorithm for DCVS [21]. Up to our best knowledge,

there also exist other literatures about CS-based video coding, e.g., [22]-[23].

B. Contribution of This Paper

In this paper we propose a novel for sampling and recovering of compressed sensed video data. The proposed method divides the video sequence into the key and non-key frames follow by dividing each frame into the small blocks of equal sizes, similar to [13]. Each blocks of key (non-key) frames are sampled using the same sensing matrix Φ_{B_K} ($\Phi_{B_{NK}}$). The compressed key frame data are reconstructed initially using MH block compressed sensing recovery [18], in order to use as the initial image in the dictionary leaning algorithm step, to obtain an adaptively learned sparsifying (ALS) basis to exploit the spatial redundancy of frame, by an iterative procedure. Also, we investigate the effectiveness of three well-known dictionary learning methods, to adopt in our scheme. The obtained ALS basis is incorporated into the optimization problem, for the whole CS frame recovery, in form of ℓ_0 quasi-norm. In this step, a split Bregman iteration (SBI) [24]-[25] based technique is employed to solve the non-convex ℓ_0 minimization, efficiently. For the recovery of non-key frames, first we initialize a prediction of current frame, $\hat{F}_t$, by means of previous reconstructed frame $\tilde{F}_{t-1}$, in order to exploit the temporal redundancy. The prediction $\hat{F}_t$ is employed into a SBI based method together with the achieved ALS basis of current frame and current CS data, to recover the current non-key frame $\tilde{F}_t$ by solving the proper minimization problem. The experimental results show the high competitive performance of our proposed method compared with the other state-of-the-art CVS techniques.

The rest of this paper is organized as follows. In Section II, we introduce an abstract framework for CS theory and SBI method. Also, we review three well-known techniques of dictionary learning, briefly. The proposed method is described in Section III. Numerical results and comparisons for our proposed method are given in Section IV and finally, Section V concludes the paper.

II. Background

This section reviews an abstract framework of CS theory and SBI algorithm, accompanied by investigation of three well-known dictionary learning methods, briefly.

A. Compressed Sensing

Suppose we wish to recover a real value finite length signal $u \in \mathbb{R}^n$ from a finite length observation $f \in \mathbb{R}^m$; so that $m \ll n$ and there is a linear projection between them

$$f_{m \times 1} = \Phi_{m \times n} u_{n \times 1} + e_{m \times 1} \tag{1}$$

where $\Phi \in \mathbb{R}^{m \times n}$ is a sensing matrix and $e \in \mathbb{R}^{m \times 1}$ denotes the additive noise. Since the number of unknowns is much more than the observations, clearly we are not able to recover every u from f and it is generally considered as an *ill-posed* problem. However, if u be sufficiently sparse in the sense that u can be written as a superposition of a small number of vectors taken from a known (sparsifying) transform domain basis ($t = n$) or frame ($t > n$) $\Psi \in \mathbb{R}^{t \times n}$ or even adaptive learned sparsifying basis (e.g., see [26]), then exact recovery of u is possible. So sparsity plays a key role in recovering of u from observation vector f. Also, u would be called s-sparse if only its s coefficients in the set of transform domain ($\vartheta = \Psi u$) are nonzero and the other $n - s$ coefficients are zero. In other words, the transform domain signal ϑ can be well approximated using only $s < m \ll n$ nonzero entries. In order to solve the reconstruction problem with a reasonable accuracy and robustness to the noise, the estimation of u is formulated as an unconstrained Lagrangian optimization problem which incorporates the prior information about the original signal

$$\min_{u}\{\frac{1}{2}\|f - \Phi u\|_{\ell_2}^2 + \lambda\|\Psi u\|_{\ell_p}\} \tag{2}$$

or

$$\min_{\vartheta}\{\frac{1}{2}\|f - \Phi\Psi^{-1}\vartheta\|_{\ell_2}^2 + \lambda\|\vartheta\|_{\ell_p}\}, \tag{3}$$

where $u = \Psi^{-1}\vartheta$. Here, the first term is a penalty that represents the closeness of the solution to the observed scene and quantifies the prediction error with respect to the measurements. The second term is a regularization term that represents a priori sparse information of the original scene and also it is designed to penalize an estimate that would not exhibit the expected properties. Also, λ is a regularization parameter that balances the contribution of both terms. In the second term of (2) or (3), ℓ_p is usually considered as ℓ_0 or ℓ_1. This minimizing problem can be solved easily by iterative shrinkage/thresholding (IST) methods (see, e.g., [27]), Bregman iterative algorithms (see, e.g., [24] and [25]). Since ℓ_0 minimization is non-convex and its solution is considered as *NP-hard*, the common method is to replace ℓ_0 quasi-norm with the ℓ_1 norm, because it is the closest convex norm to non-convex ℓ_0 quasi-norm and minimizing the ℓ_1 norm instead. However, a fact that is often neglected is, for some practical problems, i.e., image inverse problems, the conditions guaranteeing the equivalence of ℓ_0 minimization and ℓ_1 minimization are not necessarily satisfied.

B. Split Bregman Iteration

The split Bregman iteration (SBI) method was recently proposed by Goldstein and Osher [24] for effectively solving ℓ_1-regularized optimization problem with multiple ℓ_1-regularized terms [25]. The basic

TABLE I: The SBI Algorithm Framework

Input: $\{F_1(v)\}, \{F_2(w)\}$, G, $\mu > 0$

Output: v: Reconstructed signal;

Initialization: Set $k = 0$; $(v^0, w^0, b^0) = (0, 0, 0)$

While a stop criterion is not satisfied **do**

1. $\quad v^{k+1} = \arg\min_v F_1(v) + \frac{\mu}{2}\|w^k - Gv - b^k\|_{\ell_2}^2$

2. $\quad w^{k+1} = \arg\min_w F_2(w) + \frac{\mu}{2}\|w - Gv^{k+1} - b^k\|_{\ell_2}^2$

3. $\quad b^{k+1} = b^k + Gv^{k+1} - w^{k+1}$

4. $\quad k \leftarrow k + 1$

End While

Note: $w = Gv$.

---

idea of SBI is to convert the unconstrained minimization problem into a constrained one by introducing the variable splitting technique and then invoke the Bregman iteration [28] to solve the constrained minimization problem. The rationale behind SBI is that it may be easier to solve the constrained problem than to solve its unconstrained counterpart. Another advantage of the SBI is that it has relatively small memory footprint and is easy to program by users. Such properties are very attractive for large-scale problems.

Consider an unconstrained optimization problem

$$\min_{v \in \mathbb{R}^N}\{F_1(v) + F_2(Gv)\} \tag{4}$$

where $G \in \mathbb{R}^{M \times N}$, $F_1 : \mathbb{R}^N \to \mathbb{R}$ and $F_2 : \mathbb{R}^M \to \mathbb{R}$. Apparently, (4) can be converted into an equivalent constrained form and then solved by invoking the SBI. The SBI framework is presented in Table I.

C. Sparse Representation and Dictionary Learning

As stated previously, the key of the sparse representation modeling lies in the choice of sparsifying basis (or dictionary D). In this context, $D = [d_1, d_2, \cdots, d_t] \in \mathbb{R}^{n \times t}$ is called a dictionary and each of its columns is called an atom. One crucial problem in a sparse-representation problem is how to choose the efficient dictionary. There are many pre-specified (non-adaptive analytically designed) sparsifying dictionaries (basis or frame), e.g., Fourier, discrete cosine transform, wavelets, Ridgelets, Curvelets, Contourlets, Shearlets and etc. In spite of being simple and having fast computations, the analytically designed dictionaries are not able to efficiently (sparsely) represent a given class of signals, and they lack the adaptivity to the image local structures. However, learning the atoms from a set of training signals

belonging to signal class of interest would result in dictionaries with the capability of better matching the content of the signals [29]. It has been experimentally shown that these adaptive dictionaries outperform the non-adaptive ones in many signal processing applications.

Dictionary learning algorithms iteratively perform the two stages of *sparse approximation* (*sparse coding*) and *dictionary update*. In the first stage, which is actually the clustering of the signals into a union subspace, the sparse approximation of the signals is computed using the current dictionary. The second stage is the update of the dictionary. To the best of our knowledge, most dictionary learning algorithms differ mainly in the way of updating the dictionaries [30]-[32]. Some algorithms such as K-singular value decomposition (K-SVD) [31] are based on updating the atoms one-by-one, while some others such as method of optimal directions (MOD) [30] updates the whole set of atoms at once. In [32], a MOD-like algorithm was proposed in which more than one atom along with the non-zero entries in their associated row vectors in coefficient matrix are updated at a time. We refer to this algorithm as the multiple dictionary update (MDU) algorithm (for further reading see [32]).

D. Patch-based redundant sparse recovery

In literature, the basic unit of sparse representation for natural images is patch [31]. Suppose the vector representation of the original image denotes by $u \in \mathbb{R}^n$, accordingly, $u_{p_l} \in \mathbb{R}^{B_s}$ represents an image patch of size $\sqrt{B_s} \times \sqrt{B_s}$ at location l, $l = 1, 2, \cdots, J$. Then we have

$$u_{p_l} = R_l u, \tag{5}$$

where $R_l \in \mathbb{R}^{B_s \times n}$ is a binary matrix operator that extracts the square patch u_l from u, forming the output patch as the column vector. Note that, patches are usually overlapped (to suppress the boundary artifacts), and such patch based representation is highly redundant and significant to achieve high recovery quality. Therefore, the recovery of u from $\{u_{p_l}\}$ becomes an over-determind system, which is straightforward to obtain the following least-square solution:

$$u = (\sum_{l=1}^{J} R_l^T R_l)^{-1} \sum_{l=1}^{J} (R_l^T u_{p_l}), \tag{6}$$

which is nothing but telling that the overall image is reconstructed by averaging all the overlapped patches. Given dictionary D, the sparse coding process of each patch u_{p_l} over D is to find a sparse vector α_l such that $u_{p_l} \approx D\alpha_l$. Then the entire image can be sparsely represented by the set of sparse codes $\{\alpha_l\}$:

$$u \approx D \circ \alpha = (\sum_{l=1}^{J} R_l^T R_l)^{-1} \sum_{l=1}^{J} (R_l^T D\alpha_l), \tag{7}$$

where α denotes the concatenation of all $\{\alpha_l\}$. By given a set of training image patches $P = [u_{p_1}, u_{p_2}, \cdots, u_{p_{J'}}]$, where J' is the number of training image patches, the goal of sparsifying basis learning is to jointly optimize the sparsifying basis D, and the representation coefficient matrix $\Lambda = [\alpha_1, \alpha_2, \cdots, \alpha_{J'}]$; such that $u_{p_l} = D\alpha_l$ and $\|\alpha_l\|_{\ell_p} \leq L$, where ℓ_p is ℓ_0 or ℓ_1. This can be formulated by the following minimization problem:

$$(\hat{D}, \hat{\Lambda}) = \arg \min_{D, \Lambda} \sum_{l=1}^{J'} \|u_{p_l} - D\alpha_l\|_{\ell_2}^2 \quad s.t. \quad \|\alpha_l\|_{\ell_p} \leq L, \forall l, \tag{8}$$

where the requirement of $\|\alpha_l\|_{\ell_0} \leq L \ll n$ indicates that the sparse representation stage uses no more than L atoms from the dictionary for every image patch instance.

Although the above minimization problem in equation (8) is large-scale and highly non-convex even when ℓ_p is equal to ℓ_1, some approximation approaches (e.g., MOD [30], K-SVD [31] and MDU [32]) have been proposed to optimize D and Λ alternatively, leading to many state-of-the-art results in image processing.

III. THE PROPOSED METHOD

By contrast with the conventional/distributed video coding scheme, in which data acquiring and compression tasks are performed disjointedly, CVS employs both data acquiring (video sensing) and compression, into a unified task which emerges a new procedure to directly acquiring compressed video data via random projection (without temporally storing the complete raw data) for each individual frame (or blocks of frame) at a low complexity encoder. In this case, the majority of computational burden is shifted from the encoder to the decoder side, which is more suitable to deploy in modern video applications, e.g., video serveillance systems and wireless multimedia sensor networks. In this section, first we discuss about encoding of the proposed method, and then we propose decoding scheme of key and non-key frames.

A. Encoding

As mentioned before, the proposed method, firstly, divides the video sequence into the key and non-key frames follow by dividing each frame, of size $I_r \times I_c$, into small non-overlapping blocks of equal size (i.e., size $B \times B$), and then the same sensing (sampling/measurement) matrix Φ_B[1] (i.e., size $m_B \times B^2$,

---

[1]Note that, $\frac{m}{n}$ denotes the measurement ratio; the measurement ratio for key and non-key frame may differ. In this context, MR_K (MR_{NK}) and Φ_{BK} (Φ_{BNK}) are used as the measurement ratio and the sensing matrix of key (non-key) frames, respectively.

where $m_B = \lfloor \frac{mB^2}{n} \rfloor, n = I_r I_c$) is applied for sampling of eah block. In this case, we have:

$$f_i = \Phi_B u_i \tag{9}$$

where u_i is a (column) vector representing of block i of the input image, f_i is its corresponding measurement vector and Φ_B independently samples blocks within frame. Using this technique has several benefits comparing to use of a random sampling operator to the entire image; i.e., the encoder does not need to wait until the entire image is measured, but each block is sent after its linear projection. In addition, at the decoder side, each block is processed independently; therefore the speed of encoding and decoding procedure is increased. Also in this case, we just need to store a $m_B \times B^2$ sensing matrix instead of a $m \times n$ sensing matrix. More precisely, the global sensing matrix takes a block-diagonal structure, $\Phi = \mathrm{diag}(\Phi_B, \cdots, \Phi_B)$.

B. Recovery of key frame

At the decoder side, the key frame is reconstructed initially using method of the multi-hypothesis block CS recovery [11], in order to use as initial training image in the process of learning an ALS basis (dictionary). Now, in this case, by considering $u \approx D \circ \alpha$, The equation (3) using ALS basis can be written as

$$\min_{\alpha} \{ \frac{1}{2} \| f - \Phi D \circ \alpha \|_{\ell_2}^2 + \lambda \| \alpha \|_{\ell_p} \}. \tag{10}$$

Here, D replaces Ψ^{-1} in equation (3), standing for ALS basis. Also, α denotes the patch-based redundant sparse representation for the whole image over D, which can be find by solving (10). As discussed previously, when ℓ_p is replaced with ℓ_0, since ℓ_0 is non-convex and NP-hard, the usual routine is to solve its optimal convex approximation, i.e., ℓ_1 minimization. However, for some practical problems, i.e., image inverse problems, the conditions guaranteeing the equivalance of ℓ_0 minimization and ℓ_1 minimization are not necessarily satisfied. An approach was proposed in [26], where equation (10) can be efficiently solved via ALS basis and ℓ_0 minimization. In this part, we adopt the proposed scheme of [26] to solve (10), which leads to recovery of the key frames. Now, let's go back to equation (10) and point out how to solve it. By considering $v = D \circ \alpha$ and $\ell_p = \ell_0$ (sparsity is strictly measured), (10) can be formulated into an equivalent constrained form as:

$$\min_{\alpha, v} \{ \frac{1}{2} \| f - \Phi v \|_{\ell_2}^2 + \lambda \| \alpha \|_{\ell_0} \} \quad s.t. \quad v = D \circ \alpha. \tag{11}$$

In order to solve the above minimization problem, an alternating SBI algorithm (as illustrated in Table. 1) is applied. We finally obtain the following schemes:

$$v^{k+1} = \arg \min_v \frac{1}{2} \| f - \Phi v \|_{\ell_2}^2 + \frac{\mu}{2} \| D \circ \alpha^k - v - b^k \|_{\ell_2}^2, \tag{12}$$

$$\alpha^{k+1} = \arg\min_{\alpha} \lambda\|\alpha\|_{\ell_0} + \frac{\mu}{2}\|D \circ \alpha - v^{k+1} - b^k\|_{\ell_2}^2, \tag{13}$$

$$b^{k+1} = b^k + v^{k+1} - D \circ \alpha^{k+1}. \tag{14}$$

Here, μ is a fixed parameter for improving the numerical stability of the algorithm.

Given α^k, the sub-problem of (12) consists in minimizing a strictly convex quadratic function, that can be solved easily. By setting the gradient of the objective function in (12) to be zero, a closed solution for (12) is achieved (see the Appendix I), which can be expressed as

$$v^{k+1} = (\Phi^T\Phi + \mu I)^{-1}\big(\mu(D \circ \alpha^k - b^k) + \Phi^T f\big), \tag{15}$$

where $(\cdot)^{-1}$ and I denote the matrix inverse operator and the identity matrix, respectively. Since for image CS recovery and also here, Φ is a random projection matrix. Therefore, it is too costly to solve the minimization of the quadratic function in (12) directly by using (15), because of existence of the matrix inverse in (15). Here, in order to avoid computing the inversion of matrix, the steepest descent method with the optimal step is utilized to solve the minimization of the quadratic function in (12), which can be expressed as

$$v^{k+1} = v^k - \eta^k g^k. \tag{16}$$

Here, g is the gradient direction of the objective function and η represents the optimal step size; for finding η, see the Appendix I.

Now, by given v in hand (according to (12)), and by considering $r = v + b$ and $u = D \circ \alpha$ (for simplicity, the subscript k is dropped without confusion), the sub-problem of (13) becomes

$$\min_{\alpha} \frac{1}{2}\|u - r\|_{\ell_2}^2 + \frac{\lambda}{\mu}\|\alpha\|_{\ell_0}. \tag{17}$$

By these transformations, we regard r as some type of the noisy observation of u. However, it is worth to note that, due to the complicated definition of α, it is difficult to solve (17) directly. In order to solve (17) amenable, in this paper, a reasonable assumption is used, which leads to obtain a closed-form solution of (17). Such this assumption is efficiently employed in [14] and [33], hence we utilize the same approach to solve the above minimization problem.

Let denote the error vector by $e = u - r$ (such that $u, r \in \mathbb{R}^n$) and each element of e by $e(i), i = 1, \cdots, n$. Also, we suppose that each element of e $(e(i))$ follows an independent zero-mean distribution, $E\{e(i)\} = 0$, with the same variance $Var\{e(i)\} = E\{e^2(i)\} = \sigma^2$, where $Var\{\cdot\}, E\{\cdot\}$ represent the variance and the expectation operator, respectively. It is worth emphasizing that the above assumption does not need to be Gaussian, or Laplacian, or generalized Gaussian distribution process, which are

more general. By invoking the *law of large numbers* in probability theory, for any $\epsilon > 0$, it leads to $\lim_{n \to \infty} \Pr\{|\frac{1}{n} \sum_{i=1}^{n} e^2(i) - \sigma^2| < \frac{\epsilon}{2}\} = 1$, i.e.,

$$\lim_{n \to \infty} \Pr\{\left|\frac{1}{n}\|u - r\|_{\ell_2}^2 - \sigma^2\right| < \frac{\epsilon}{2}\} = 1, \tag{18}$$

where $\Pr\{\cdot\}$ represents the probability.

Let u_c, r_c denote the concatenation of all the patches u_{p_l} and r_{p_l}, $l = 1, 2, \cdots, J$, respectively, and each element of $u_c - r_c$ is denoted by $e_c(j), j = 1, \cdots, K$, where $K = B_s \times J$. In accordance with the assumption, it is concluded that $e_c(j)$ is independent with zero mean and variance σ^2. Thus, due to the *law of the large numbers* and by doing the same manipulation with (18) to $e_c^2(j)$, it yields $\lim_{K \to \infty} \Pr\{|\frac{1}{K} \sum_{j=1}^{K} e_c^2(j) - \sigma^2| < \frac{\epsilon}{2}\} = 1$, i.e.,

$$\lim_{K \to \infty} \Pr\{\left|\frac{1}{K} \sum_{l=1}^{J} \|u_{p_l} - r_{p_l}\|_{\ell_2}^2 - \sigma^2\right| < \frac{\epsilon}{2}\} = 1. \tag{19}$$

Therefore, according to (18) and (19), the following property is concluded

$$\lim_{\substack{n \to \infty \\ K \to \infty}} \Pr\{\left|\frac{1}{n}\|u - r\|_{\ell_2}^2 - \frac{1}{K} \sum_{l=1}^{J} \|u_{p_l} - r_{p_l}\|_{\ell_2}^2\right| < \epsilon\} = 1, \tag{20}$$

in which, the relationship between $\|u - r\|_{\ell_2}^2$ and $\sum_{l=1}^{J} \|u_{p_l} - r_{p_l}\|_{\ell_2}^2$ (with large probability) is described:

$$\frac{1}{n}\|u - r\|_{\ell_2}^2 = \frac{1}{K} \sum_{l=1}^{J} \|u_{p_l} - r_{p_l}\|_{\ell_2}^2. \tag{21}$$

Now, by substituting (21) into (17), J sub-problems for all the patches u_{p_l} is achieved that they can be solved more efficiently. Each patch based sub-problem is formulated as

$$\arg\min_{\alpha_l} \frac{1}{2}\|u_{p_l} - r_{p_l}\|_{\ell_2}^2 + \theta\|\alpha_l\|_{\ell_0}, \tag{22}$$

where $\theta = \frac{\lambda K}{\mu n}$. In fact, by considering $u_{p_l} = D\alpha_l$, where D is the adaptive learned dictionary from r_{p_l} using patch-based dictionary learning method described in Section II-D, the above sub-problem can be considered as the sparse coding problem, i.e.,

$$\arg\min_{\alpha_l} \frac{1}{2}\|D\alpha_l - r_{p_l}\|_{\ell_2}^2 + \theta\|\alpha_l\|_{\ell_0}. \tag{23}$$

Also, in order to achieve higher sparsity, (23) can be formulated in its constrained form, i.e.,

$$\min_{\alpha_l} \|\alpha_l\|_{\ell_0} \quad s.t. \quad \|D\alpha_l - r_{p_l}\|_{\ell_2}^2 \le \delta \tag{24}$$

where $\delta = \omega\theta$ is a small constant controlling the approximation error, and ω is a control factor. Now, (24) can be solved efficiently using orthogonal matching pursuit (OMP) [34] algorithm. However, if ℓ_0 pseudo-norm in (24) was relaxed with ℓ_1 convex norm, it could be solved with basis pursuit [35],

TABLE II: THE DETAILED DESCRIPTION OF KEY FRAME RECOVERY FRAMEWORK

Input: f, $\Phi_B, B_s, \omega, \lambda, iin$: inloop iteration number

k_{max}: maximum iteration number, Tol: Tolerance

Output: $u^* = D \circ \alpha^*$: Recovered key frame;

Initialization: Set $k = 0$; $(\alpha^0, b^0) = (0, 0)$

$u_{init} = v^0 =$ MH_recovery(f, Φ_B) using method of [18]

While a stop criterion is not satisfied **do**

1. $v^{k+1} = \tilde{v} = v^k$

     $r^{k+1} = v^{k+1} + b^k$

2.   update D^{k+1} using (25)

3.   **for** each patch u_{p_l} **do**

     compute α_l^{k+1} using (24)

     end for

4.   update α^{k+1} by concatenating all $\{\alpha_l^{k+1}\}$

5.   **for** $i = 1 : iin$

     $g_i^k \leftarrow \Phi^T \Phi v^k - \Phi^T f - \mu(D^{k+1} \circ \alpha^{k+1} - v^{k+1} - b^k)$

     $\eta_i^k \leftarrow \mathrm{diag}\big(\mathrm{abs}(\frac{g_i^{k\,T} g_i^k}{g_i^{k\,T}(\Phi^T\Phi + \mu I)g_i^k})\big)$

     $v^{k+1} \leftarrow v^{k+1} - \eta_i^k g_i^k$

     end for

6.   compute $s^{k+1} =$ SSIM$(v^{k+1}, \tilde{v})$

7.   compute $diff =$ abs$(s^{k+1} - s^k)$

8.   update b^k using (14)

9. $k \leftarrow k + 1$

end While

stopping criterion: $k = k_{max}$ or $diff \leq Tol$.

lasso [36] and ℓ_1-regularized least square [37], but may be at the cost of less sparse solution. For all J overlapped patches, this process is employed to achieve α (concatenation of all $\{\alpha_l\}, l = 1, \cdots, J$), which is the solution of the sub-problem (17).

Thus, the key frame is recovered efficiently, by solving the optimization problem of (11) using discussed SBI method (e.g., see (12)-(14)) via updating dictionary using (8) in which, u_{p_l} is replaced by r_l, (since r is regarded as a good approximation of u at each iteration, here, we conduct adaptive sparsifying basis learning using all the patches extracted from r),

$$\hat{D} = \arg\min_{D} \sum_{l=1}^{J'} \|r_{p_l} - D\alpha_l\|_{\ell_2}^2 \quad s.t. \quad \|\alpha_l\|_{\ell_0} \leq L, \forall l. \tag{25}$$

The proposed algorithm for recovery of key frame data is described in detail in Table II.

C. Recovery of non-key frame

While spatial domain compression is performed by CS, the temporal redundancy is not exploited fully, because no motion estimation and compensation is performed at the CVS encoder. To incorporate the temporal redundancy, in order to efficient recovery of the non-key frames, the temporal correlation between adjacent frames is exploited through the inter-frame sparsity model. Here, an iterative approach for the reconstruction of the non-key frames is adopted, where the approach initially estimates an approximation of the non-key frame using previous reconstructed frame, in order to take the advantage of the inherent temporal structure between successive frames. Then, the initially estimated frame is utilized into an optimization problem to recover and refine the current non-key frame.

Assume that $u_{t-1}^* = D \circ \alpha_{t-1}^*$, where u_{t-1}^* is the previous reconstructed frame using ALS basis D, and α_{t-1}^* denotes the patch-based redundant sparse representation for the whole previous reconstructed frame over D. The initialization step can be formulated as:

$$\min_{\alpha} \{ \frac{1}{2} \| f - \Phi D \circ \alpha \|_{\ell_2}^2 + \lambda \| \alpha \|_{\ell_1} + \tau \| \alpha - \alpha_{t-1}^* \|_{\ell_1} \}, \tag{26}$$

where the first term keeps the solution close to the measurements, the second term promotes sparsity in the spatial transform of the current frame, and the third term promotes sparsity in the inter-frame difference to achieve the temporal redundancy between the current frame and the previous reconstructed frame. In this paper, the problem of (26) is solved efficiently by an alternating SBI-based framework via an iterative-shrinkage method using surrogate function.

Assume that $v = D \circ \alpha$, then (26) can be formulated in its constrained form:

$$\min_{\alpha, v} \{ \frac{1}{2} \| f - \Phi v \|_{\ell_2}^2 + \lambda \| \alpha \|_{\ell_1} + \tau \| \alpha - \alpha_{t-1}^* \|_{\ell_1} \} \quad s.t. \quad v = D \circ \alpha. \tag{27}$$

Now, the above equation can be solved easily by an alternating SBI-based framework as follow:

$$v^{k+1} = \arg\min_v \{ \frac{1}{2} \| f^k - \Phi v \|_{\ell_2}^2 + \frac{\mu}{2} \| D \circ \alpha^k - v - b^k \|_{\ell_2}^2 \},$$

$$\alpha^{k+1} = \arg\min_\alpha \{ \frac{\mu}{2} \| D \circ \alpha - v^{k+1} - b^k \|_{\ell_2}^2 + \lambda \| \alpha \|_{\ell_1} + \tau \| \alpha - \alpha_{t-1}^* \|_{\ell_1} \},$$

$$b^{k+1} = b^k + v^{k+1} - D \circ \alpha^{k+1},$$

$$f^{k+1} = f^k + f - \Phi v^{k+1}. \tag{28}$$

The last term in (28), adds a residual feedback to the algorithm.

Given α^k, b^k and f^k, the first sub-problem in (28) is essentially a minimization problem of strictly convex quadratic function, $Q_2(v)$, that is

$$\min Q_2(v) = \min_v \{ \frac{1}{2} \| f^k - \Phi v \|_{\ell_2}^2 + \frac{\mu}{2} \| D \circ \alpha^k - v - b^k \|_{\ell_2}^2 \}. \tag{29}$$

Setting the gradient of the objective function in (29) to be zero, leads to a closed solution resemble to that expressed in (15), but, here f is replaced with f^k. Resemble (15), here exists matrix inverse that is too costly to compute, when Φ is a random projection matrix. Therefore, again the steepest descent method is used to solve (29), efficiently.

By given v^{k+1}, and assuming $u = D \circ \alpha$ and $r = v + b$ (for the simplicity the subscript k is omitted without confusion), the α sub-problem in (28) can be formulated as

$$\min_{\alpha} \left\{ \frac{1}{2} \|u - r\|_{\ell_2}^2 + \frac{\lambda}{\mu} \|\alpha\|_{\ell_1} + \frac{\tau}{\mu} \|\alpha - \alpha_{t-1}^*\|_{\ell_1} \right\}. \tag{31}$$

Now, by utilizing the same assumption which used in Section III-B, the sub-problem of (31) can be formulated as follow

$$\min_{\alpha} \sum_{l=1}^{J} \left\{ \frac{1}{2} \|u_{p_l} - r_{p_l}\|_{\ell_2}^2 + \frac{\lambda K}{\mu n} \|\alpha_l\|_{\ell_1} + \frac{\tau K}{\mu n} \|\alpha_l - \alpha_{t-l_l}^*\|_{\ell_1} \right\}. \tag{32}$$

Obviously, (32) can be solved efficiently by solving J sub-problems for all patches. By assuming $u_{p_l} = D\alpha_l$, $\theta_1 = \frac{\lambda K}{\mu n}$, $\theta_2 = \frac{\tau K}{\mu n}$ and for a single patch r_{p_l}, we have

$$\arg\min_{\alpha_l} \left\{ \frac{1}{2} \|D\alpha_l - r_{p_l}\|_{\ell_2}^2 + \theta_{1,l} \|\alpha_l\|_{\ell_1} + \theta_{2,l} \|\alpha_l - \alpha_{t-l_l}^*\|_{\ell_1} \right\}, \tag{33}$$

where $\theta_\chi = \mathrm{diag}\left((\theta_{\chi,l})_{l=1}^J \right), \chi = \{1, 2\}$.

The sub-problem of (33) can be solved adeptly via an iterative-shrinkage algorithm using surrogate function (see Appendix II, for details). Such this method is based on the work of Daubechies *et al*. [38], and was used proficiently by Dong *et al*. [39] in order to solve a double-header ℓ_1 optimization.

We utilize the solution of (27) into an refinement (post processing) optimization problem in order to enhance the recovery of current frame. Suppose v^* is the solution of (27), then the refinement optimization problem can be formulated as:

$$\{u_t^*, \{\alpha_{t_l}^*\}_{l=1}^J, D\} = \min_{\alpha_t, u_t, D_t} \left\{ \frac{1}{2} \|f - \Phi u_t\|_{\ell_2}^2 + \frac{1}{2} \|u_t - v^*\|_{\ell_2}^2 + \sum_{l=1}^{J} \mu_l' \|\alpha_{t_l}\|_{\ell_0} + \frac{\lambda'}{2} \sum_{l=1}^{J} \|D_t\alpha_l - R_l u_t\|_{\ell_q} \right\}, \tag{34}$$

where $\mu_l > 0$ are some regularization parameters that control the image patch sparsity. Besides, λ' is weight parameter controls the trade-off between the data fidelity and the image prior. Here, D_t is the ALS basis of current frame that can be learned (initially using D and v^*) and updated via the methods of [30]-[32]. Indeed, the second term measures the distance between the estimated frame and the enhanced one. The problem of (34) can be solved in an iterative mode, by decoupling it into three sub-problems of *sparse coding*, *dictionary learning*, and *reconstruction*. The sparse coding problem can be solved using

OMP, and D_t, in dictionary learning step, updates by the methods of [30]-[32]. Also, the reconstruction step has a closed-form solution

$$u_t^{k+1} = (\Phi\Phi^T + I + \lambda' \sum_{l=1}^{J} R_l^T R_l)^{-1}(v^* + \Phi^T f + \lambda' \sum_{l=1}^{J} R_l^T D\alpha_l). \tag{35}$$

At a glance, it seems that (34) is very similar to that proposed in [40], however, in fact we use (34) in order to update the variables which used for the recovery of next frame, additionally with a minor enhancement on the recovered frame using (27).

IV. EPERIMENTAL RESULTS

In this section, we evaluate the performance of the proposed method. To evaluate our simulation results, we use two applicable quality assessors, the peak signal-to-noise (PSNR) in dB and the structural similarity (SSIM) [41]. For an 8 bits gray scale $I_r \times I_c$ image (frame), PSNR is calculated as

$$PSNR = 20 \log_{10} \frac{\sqrt{I_r \times I_c} 255}{\|u - \tilde{u}\|}, \tag{36}$$

where u and $\tilde{u}$ denote the original image and the reconstructed image, respectively.

SSIM [41] is another assessor for measuring the similarity between two images, and it is near to human eye perception. Here, the obtained results are evaluated by a specific form of the SSIM index:

$$SSIM(u, \tilde{u}) = \frac{(2\mu_u\mu_{\tilde{u}} + c_1)(2\sigma_{u\tilde{u}} + c_2)}{(\mu_u^2 + \mu_{\tilde{u}}^2 + c_1)(\sigma_u^2 + \sigma_{\tilde{u}}^2 + c_2)} \tag{37}$$

where μ_u ($\mu_{\tilde{u}}$) and σ_u ($\sigma_{\tilde{u}}$) denote the average and variance of u ($\tilde{u}$), respectively. $\sigma_{u\tilde{u}}$ is the covariance of u and $\tilde{u}$. c_1 and c_2 are two variables to stabilize the division with weak denominator, such that $c_1 = (k_1 L)^2, c_2 = (k_2 L)^2$ and $0 < k_1, k_2 \ll 1$. L is the dynamic range of pixel values (255 for 8 bits gray scale images). For our experiments, we use $k_1 = 0.01$ and $k_2 = 0.03$. All experiments were performed using MATLAB 2013a, on a computer equipped with Intel® core^{TM} i7, 3.7 GHz processor, with 48 GB of RAM, and running on Windows 7. The performances of our experiments are evaluated on the luminance component of four well-known video test sequences (e.g., "*Foreman*", "*Coastguard*", "*Mobile and Calendar*" and "*Hall Monitor*") with a CIF resolution of 352×288 pixels. Also, in all experiments we use the block size of 32×32 and the size of each patch is set to 8×8. The CS measurement of each blocks is obtained by applying an orthonormalized i.i.d. Gaussian projection matrix to each of them, however the orthonormality of the projection matrix makes the solution of problems more simpler. Note that, in this paper we do not consider the quantization and entropy encoding of measurements, since they are beyond the scope of this paper.
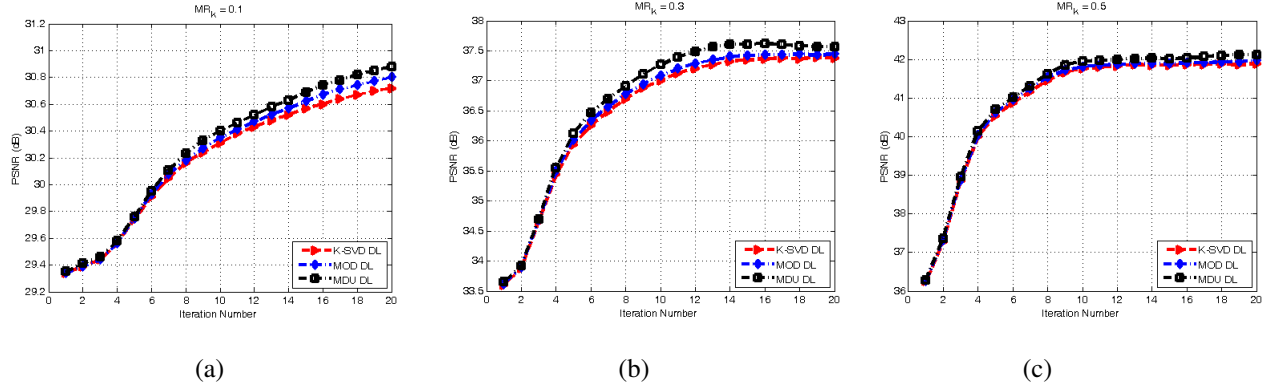
Fig. 1: Performance of different dictionary learning algorithms for the proposed method, on the frame 21st of *Foreman* sequence with various MR_K. (a) $MR_K = 0.1$: $\bar{t}_{K-SVD} = 11.41$ sec, $\bar{t}_{MOD} = 12.57$ sec, $\bar{t}_{MDU} = 112.33$ sec. (b) $MR_K = 0.3$: $\bar{t}_{K-SVD} = 15.1$ sec, $\bar{t}_{MOD} = 16.05$ sec, $\bar{t}_{MDU} = 169.8$ sec. (c) $MR_K = 0.5$: $\bar{t}_{K-SVD} = 16.96$ sec, $\bar{t}_{MOD} = 17.78$ sec, $\bar{t}_{MDU} = 203.25$ sec.

A. Experiment 1

In the first experiment, we evaluate the effectiveness of the discussed dictionary learning methods of [30]-[32] for the proposed key frame recovery algorithm. Though, the other dictionary learning methods can be investigated to adopt in our scheme. The key frame is reconstructed initially using the method of [18], in order to use as the initial training image in the process of learning an ALS basis (dictionary). The ALS basis is obtained by each of the methods provided in [30]-[32], in which the default parameter setting is as follows: the size of sparsifying basis (dictionary) is 256 and number of training iteration is 20. Also, in corresponding recovery problems (see Table II), $iin = 200$, $k_{max} = 20$, $Tol = 10^{-4}$, μ is set to be 2.5×10^{-3} and λ is set empirically. In Fig. 1, a twenty-time-iteration of our method is illustrated as an example to show the performance of MOD [30], K-SVD [31] and MDU [32] as the dictionary learning methods, for recovery of the 21st frame of the *Foreman* with different MR_K. In Fig. 1, $\bar{t}$ shows the average time of dictionary learning process at each iteration. Obviously, the results show that MDU provides a better recovery performance (in quality) compared to the other mentioned methods, but with the cost of higher computational complexity. The extra cost is only generated from the dictionary update step in each iteration. Also, the PSNR performance of K-SVD and MOD are very close to each other, but the first one gains lower computational complexity. So, it seems reasonable to adopt K-SVD as the dictionary learning algorithm in our method. The same experiments were run on *Coastguard*, *Hall Monitor* and *Mobile and Calendar* sequences (with fast motion scene); the obtained
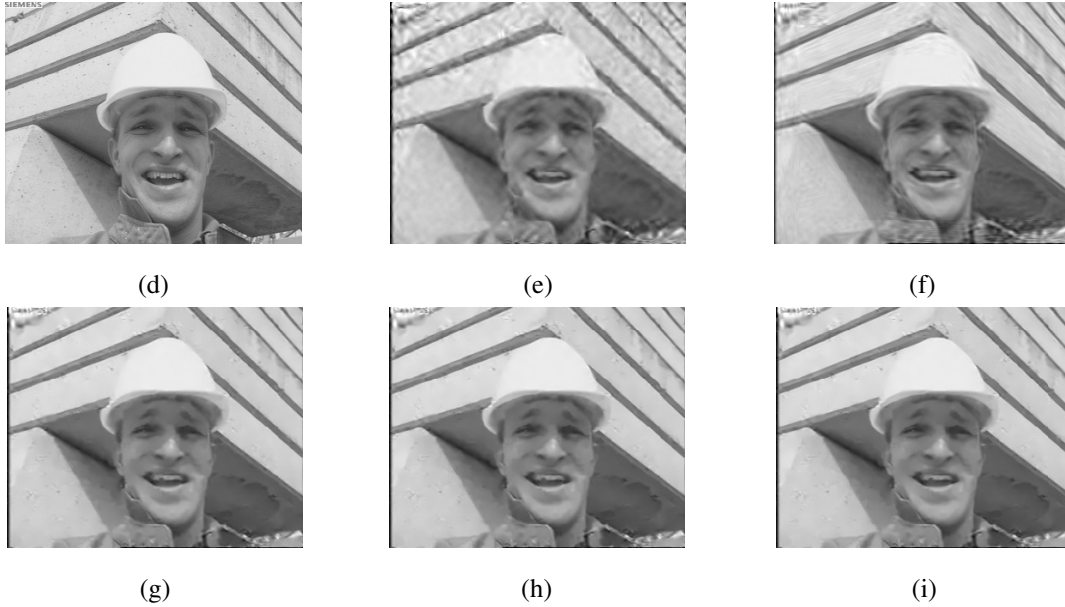
Fig. 2: Different decodings of the 21*st* frame of *Foreman* with respect to recovery as a key frame ($MR_K = 0.1$). (a) Original frame. Reconstructed 21*st* frame by usnig (b) The 2D DDWT basis (PSNR=26.82 dB, SSIM=0.783). (c) The multi-hypothesie method [11] (PSNR=28.93 dB, SSIM=0.824). The proposed method via: (d) K-SVD (PSNR=30.72 dB, SSIM=0.875). (e) MOD (PSNR=30.8 dB, SSIM=0.876). (f) MDU (PSNR=30.88 dB, SSIM=0.878).

results proved the accuracy of our assumption. Fig. 2 shows the different decoding of the 21*st* frame of the *Foreman*, produced by 2D DDWT [42] basis intra-frame decoder (with 3 level of decomposition) [Fig. 2(b)], the MH [18] decoder [Fig. 2(c)] and the proposed key frame recovery method using K-SVD [Fig. 2(d)], MOD [Fig. 2(e)] and MDU [Fig. 2(f)]. Note that, for fair comparison, the same test conditions (i.e., the same sensing matrix) are used in all experiments and all experimental results are averaged over 5 independent trials. Evidently, it can be observed that the fixed basis intra-frame decoder and MH-based decoder suffer noticeable performance loss over the whole image, while the proposed key frame recovery decoder demonstrates a considerable reconstruction quality improvement.

B. Experiment 2

In this experiment, we evaluate the performance of the proposed method (via K-SVD dictionary learning) for decoding of the first 50 frames of the test video sequences, with different measurement ratio MR scenario. More specifically, here, two proposed CVS decoders are examined for all four test sequences, i.e., when $MR_k = MR_{NK}$, and when $MR_{NK} \leq MR_K = 0.5$. Also, for comparision purposes, two existing typical CVS decoders is considered, e.g., the 2D DDWT basis intra-frame decoder, and the
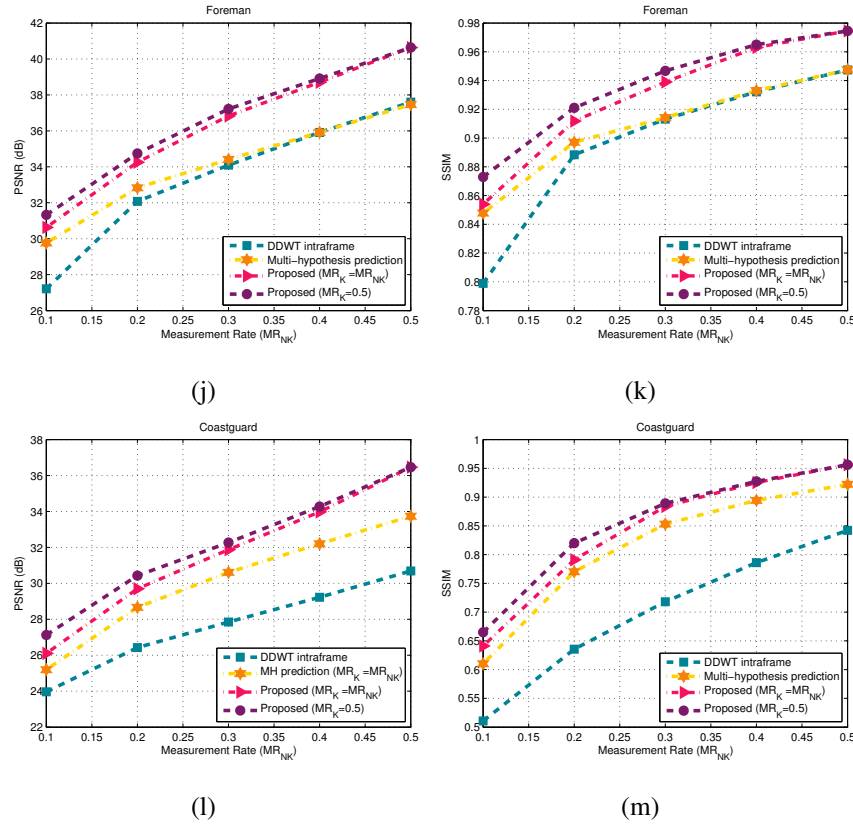
Fig. 3: The PSNR and SSIM performance of proposed method on the first 50 frames of the *Foreman* and the *Coastguard* sequence. *Foreman*: (a) PSNR vs. MR. (b) SSIM vs. MR. *Coastguard*: (c) PSNR vs. MR. (d) SSIM vs. MR.

MH predictions inter-frame decoder [18]. The group of the picture (GOP) size is set to 5 and the default parameter setting is as well as experiment 1. The number of iterations in algorithm, k_{max}, is set to 6; it is clear that the higher number of iteration yields slightly better performance in quality of recovered sequence, but at the cost of higher execution times of the algorithm. It is worth emphasizing that, however by increasing the number of iterations, the quality of recovered frame increases, but after some iterations it yields an insignificant improvement, i.e., see Fig. 1(c) that PSNR values after the 10*th* iteration converge.

In this experimental studies, Fig. 3 shows the PSNR and SSIM performance of the proposed method in various measurement ratios, compared with the other mentioned recovery algorithms for the *Foreman* and the *Coastguard* video sequences. The PSNR and SSIM values shown in Fig. 3, are averaged over all PSNR and SSIM values of the reconstructed frames. Fig. 3(a) and Fig. 3(b) show the PSNR and SSIM performance of the proposed method, respectively, compared with the 2D DDWT intra-frame and
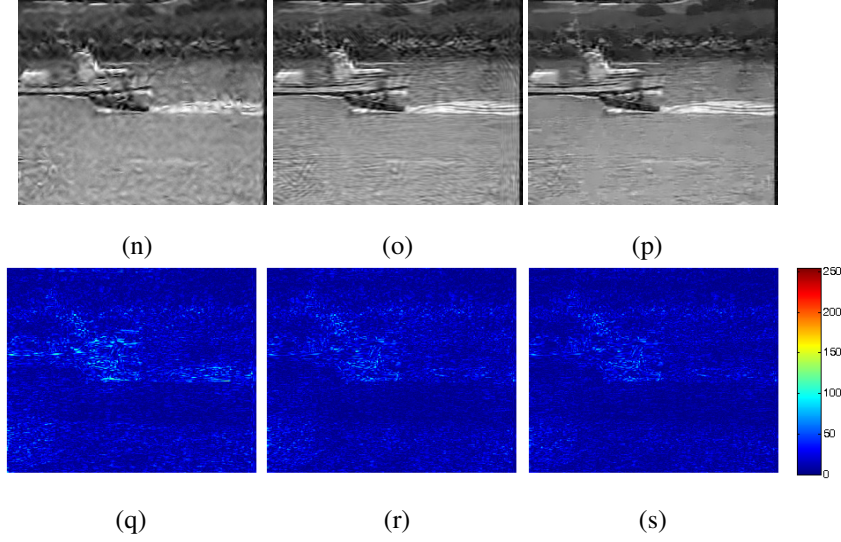
Fig. 4: Different decoding of the 49*th* frame of *Coastguard* (w.r.t. recovery as a non-key frame $MR_K = 0.5, MR_{NK} = 0.3$). (a) Using the 2D DDWT basis intra-frame decoder (PSNR=23.02 dB, SSIM=0.518). (b) Using the MH method [18] (PSNR=24.38 dB, SSIM=0.630). (c) Using the proposed method (PSNR=25.30 dB, SSIM=0.659). (d)-(f) Magnitude of error reconstruction for (a)-(c), respectively.

the MH inter-frame recovery method ($MR_K = MR_{NK}$), in various measurement ratios. Figs. 3(a)-(b) are obtained for the *Foreman* sequence, while Figs. 3(c)-(d) show the same scenario as used in Figs. 3(a)-(b), but for the *Coastguard* sequence. As can be seen, the proposed method outperforms significantly in both PSNR and SSIM values, compared to the fixed basis intra-frame and MH inter-frame decoders. Figs. 4 and 5 show the decoding comparisions with different measurement ratios scenario for the 49*th* frame of the *Coastguard*, 40*th* frame of the *Hall Monitor*, and 30*th* frame of the *Mobile and Calendar*, respectively. The frames used in Figs. 4 and 5 are all regarded as the non-key frames. Figs. 4 and 5 (d)-(f) depict the magnitude of error reconstruction (error map). Differences among the error maps are clearer with zoom-in. The error maps show the lower errors obtained by the proposed method compared to the other mentioned methods. Also, the averaged PSNR and SSIM values of all first 50 frames of the *Mobile and Calendar* and the *Hall Monitor* are listed in details in Table III.

Obviously, the numerical results show that, the proposed method (in both states, $MR_K = MR_{NK}$ and $MR_{NK} \leq MR_K = 0.5$) gains better performance, in both PSNR and SSIM, compared with the other mentioned methods. Also note that, by increasing MR_K, the reconstruction quality will increase, but poor compression is its consequence.
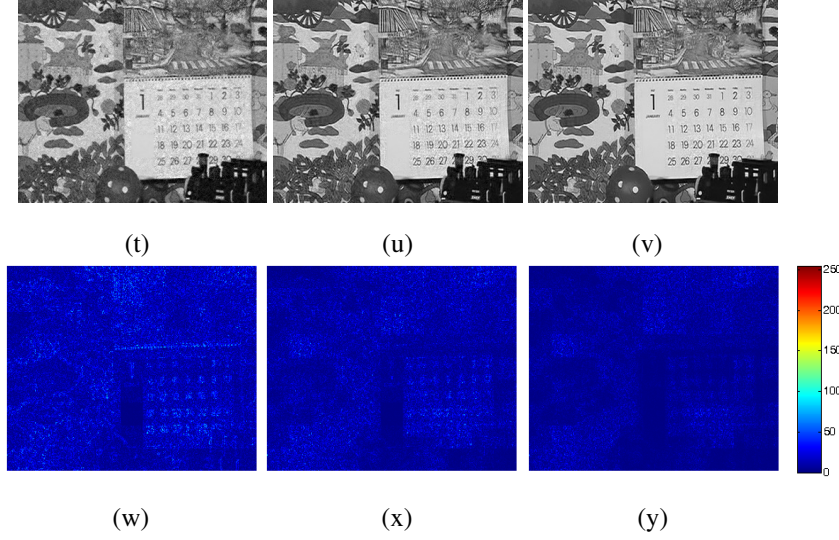
Fig. 5: Different decoding of the 30*th* frame of *Mobile and Calendar* (w.r.t. recovery as a non-key frame $MR_K = 0.4, MR_{NK} = 0.4$). (a) Using the 2D DDWT basis intra-frame decoder (PSNR=22.02 dB, SSIM=0.645). (b) Using the MH method [18] (PSNR=24.1 dB, SSIM=0.749). (c) Using the proposed method (PSNR=26.6 dB, SSIM=0.863). (d)-(f) Magnitude of error reconstruction for (a)-(c), respectively.

V. CONCLUSION

The motivation of this paper is to propose a novel framework for CVS. In our proposed method, we employed the ALS basis and ℓ_0 minimization for recovering of key frames, while non-key frames are recovered by firstly initialing a prediction of current non-key frame using previous reconstructed frame (in order to exploit the temporal redundancy), and then adopting the prediction into a proper optimization problem. Also, we investigated the effectiveness of three well-known dictionary learning algorithms (in order to learning an ALS basis), and we found out that MDU provides a better recovery performance (in quality) compared to the K-SVD and MOD, but at the cost of higher computational complexity. We found it reasonable to use the K-SVD as the dictionary learning algorithm in our proposed scheme. The numerical results show the adequacy of our proposed method in CVS, compared to the mentioned methods.

APPENDIX I

Consider the following minimization of the quadratic function:

$$\min_v Q_1(v) = \min_v \frac{1}{2}\|f - \Phi v\|_{\ell_2}^2 + \frac{\mu}{2}\|D\alpha^k - v - b^k\|_{\ell_2}^2, \tag{38}$$

TABLE III: PERFORMANCE COMPARISION IN PSNR *(dB)* AND SSIM VALUES BY DIFFERENT METHODS FOR THE *Mobile and Calendar* AND THE *Hall Monitor* VIDEO SEQUENCES

	non-key frame measurement ratio (MR$_{NK}$):				
	0.1	0.2	0.3	0.4	0.5
Algorithm	Mobile and Calendar (CIF) MR$_K$=MR$_{NK}$				
intra-frame 2D DDWT	17.570	19.389	20.687	21.948	23.274
	0.383	0.495	0.575	0.644	0.703
inter-frame MH [18]	17.548	20.359	22.243	23.9754	26.047
	0.389	0.567	0.668	0.745	0.814
Proposed method	17.877	21.181	23.812	26.461	29.463
	0.435	0.647	0.772	0.862	0.924
Algorithm	Mobile and Calendar (CIF) MR$_K$=0.5				
intra-frame 2D DDWT	18.727	20.189	21.214	22.217	23.274
	0.448	0.536	0.600	0.655	0.703
inter-frame MH [18]	19.255	21.518	23.001	24.355	26.047
	0.475	0.618	0.698	0.759	0.814
Proposed method	20.180	22.875	24.938	27.009	29.463
	0.534	0.708	0.807	0.879	0.924
Algorithm	Halll Monitor (CIF) MR$_K$=MR$_{NK}$				
intra-frame 2D DDWT	23.528	28.893	32.127	34.722	37.082
	0.761	0.888	0.927	0.949	0.963
inter-frame MH [18]	25.291	29.608	32.597	34.953	37.209
	0.814	0.896	0.930	0.947	0.961
Proposed method	26.133	31.428	35.414	38.765	41.379
	0.845	0.926	0.953	0.970	0.979
Algorithm	Hall Monitor (CIF) MR$_K$=0.5				
intra-frame 2D DDWT	26.235	30.530	33.109	35.194	37.082
	0.849	0.919	0.934	0.952	0.963
inter-frame MH [18]	27.525	31.003	33.419	35.304	37.209
	0.867	0.921	0.936	0.952	0.961
Proposed method	28.983	33.268	36.447	39.189	41.379
	0.890	0.935	0.959	0.972	0.979

that can be solved by setting its gradient to be zero:

$$\nabla Q_1 = g_1^k = -\Phi^T(f - \Phi v) - \mu(D\alpha^k - v - b^k)$$

$$= -\Phi^T f + \Phi^T \Phi v - \mu(D\alpha^k - b^k) + \mu v = 0$$

$$\Rightarrow (\Phi^T \Phi + \mu I)v = \mu(D\alpha^k - b^k) + \Phi^T f$$

$$\Rightarrow v = v^{k+1} = (\Phi^T\Phi + \mu I)^{-1}\big(\mu(D\alpha^k - b^k) + \Phi^T f\big).$$

In order to avoid using matrix inverse, the gradient descent method is utilized, i.e.,

$$v^{k+1} = v^k + \eta_1^k g_1^k, \quad \eta_1^k > 0. \tag{39}$$

By incorporating (39) into (38), and setting its partial derivative with respect to η_1^k to be zero, the optimal step size, η_1^k, is yielded, i.e., (for simplicity, the subscript k in η_1^k and g_1^k is omitted without confusion)

$$\frac{\partial Q_1}{\partial \eta_1} = g_1^T\Phi^T(f - \Phi v + \Phi\eta_1 g_1) + \mu g_1^T(D\alpha^k - v + \eta_1 g_1 - b^k) = 0,$$

$$\Rightarrow g_1^T\Phi^T\Phi\eta_1 g_1 + \mu g_1^T\eta_1 g_1 = -g_1^T\Phi^T(f - \Phi v) - \mu g_1^T(D\alpha^k - v - b^k),$$

$$\Rightarrow g_1^T(\Phi^T\Phi + \mu I)\eta_1 g_1 = g_1^T\big(\underbrace{-\Phi^T(f - \Phi v) - \mu(D\alpha^k - v - b^k)}_{g_1}\big)$$

$$\Rightarrow \eta_1 = \mathrm{abs}\big(\frac{g_1^T g_1}{g_1^T(\Phi^T\Phi + \mu I)g_1}\big).$$

APPENDIX II

The objective function in (33) is given by

$$f(\alpha, \alpha_{t-1_l}) = \frac{1}{2}\|D\alpha_l - r_{p_l}\|_{\ell_2}^2 + \theta_{1,l}\|\alpha_l\|_{\ell_1} + \theta_{2,l}\|\alpha_l - \alpha_{t-1_l}^*\|_{\ell_1},)$$

Let us add to it the following term

$$\mathrm{dist}(\alpha_l, \alpha_0) = \frac{c}{2}\|\alpha_l - \alpha_0\|_{\ell_2}^2 - \frac{1}{2}\|D\alpha_l - D\alpha_0\|_{\ell_2}^2,$$

where the parameter c is chosen in such a way that the function $\mathrm{dist}(\cdot, \cdot)$ is strictly convex (with respect to α_l), implying that its Hessian should be positive-definite, $cI - D^T D > 0$. This is satisfied by the choice $c > \|D^T D\|_{\ell_2} = \lambda_{\max}(D^T D)$ (the maximal eigenvalue of the matrix $D^T D$). The rationale behind this manipulation is to convert the objective function to a new one, for which we are able to get a closed-form expression for its global minimizer. The new objective function in (40) is called a surrogate function

$$\tilde{f}(\alpha, \alpha_{t-1_l}, \alpha_0) = \frac{1}{2}\|D\alpha_l - r_{p_l}\|_{\ell_2}^2 + \theta_{1,l}\|\alpha_l\|_{\ell_1} + \theta_{2,l}\|\alpha_l - \alpha_{t-1_l}^*\|_{\ell_1} + \frac{c}{2}\|\alpha_l - \alpha_0\|_{\ell_2}^2 - \frac{1}{2}\|D\alpha_l - D\alpha_0\|_{\ell_2}^2. \tag{40}$$

The above function can be simplified as follow

$$\tilde{f}(\alpha, \alpha_{t-1_l}, \alpha_0) = \frac{1}{2}\|D\alpha_l\|_{\ell_2}^2 + \frac{1}{2}\|r_{p_l}\|_{\ell_2}^2 - \alpha_l^T D^T r_{p_l} + \theta_{1,l}\|\alpha_l\|_{\ell_1}$$

$$+ \theta_{2,l}\|\alpha_l - \alpha_{t-1_l}^*\|_{\ell_1} + \frac{c}{2}\|\alpha_l\|_{\ell_2}^2 + \frac{c}{2}\|\alpha_0\|_{\ell_2}^2 - c\alpha_l^T\alpha_0$$

$$- \frac{1}{2}\|D\alpha_l\|_{\ell_2}^2 - \frac{1}{2}\|D\alpha_0\|_{\ell_2}^2 + \alpha_l^T D^T D\alpha_0$$

and by doing more simplification, it yields

$$\tilde{f}(\alpha, \alpha_{t-1_l}, \alpha_0) = \text{const}_1 + \theta_{1,l}\|\alpha_l\|_{\ell_1} + \theta_{2,l}\|\alpha_l - \alpha_{t-1_l}^*\|_{\ell_1} + \frac{c}{2}\|\alpha_l\|_{\ell_2}^2 - c\alpha_l^T \mathrm{v}_0, \tag{41}$$

where $\mathrm{v}_0 = \frac{1}{c}D^T(r_{p_l} - D\alpha_0) + \alpha_0$.

As can be seen, the term $\|D\alpha_l\|_{\ell_2}^2$ drops in the new function that makes the minimization much simpler. Also, the term const_1 in (41) implying all the terms that are dependent on r_{p_l} and α_0 alone.

By simplifying and reorganizing (41), the surrogate objective function can be written as

$$\tilde{f}(\alpha, \alpha_{t-1_l}, \alpha_0) = \text{const}_2 + \frac{\theta_{1,l}}{c}\|\alpha_l\|_{\ell_1} + \frac{\theta_{2,l}}{c}\|\alpha_l - \alpha_{t-1_l}^*\|_{\ell_1} + \frac{1}{2}\|\alpha_l - \alpha_0\|_{\ell_2}^2.$$

The minimizer of the above function can be obtained by an iterative-shrinkage procedure, producing sequence of results $\{\alpha_{l_i}^{k'}\}_i$ (the subscript i denotes the ith entry in α_l, and the subscript k' implies the iteration number), where at the $k+1th$ iteration, $\tilde{f}$ (and of course f) is minimized with the assignment $\alpha_0 = \alpha_l^k$ as follow

$$\alpha_{l_i}^{k'+1} = \text{sgn}(\alpha_{t+1_{l_i}}^*)\mathcal{S}_{\theta_1',\theta_2',\alpha_{t+1_{l_i}}^*}\left(\text{sgn}(\alpha_{t+1_{l_i}}^*)\mathrm{v}_i^{k'}\right),$$

where

$$\theta_\chi' = \frac{\theta_{\chi,l}}{c}, \chi = \{1,2\},$$

$$\mathrm{v}^{k'} = \frac{1}{c}D^T(r_{p_l} - D\alpha_l^{k'}) + \alpha_l^{k'},$$

and the generalized shrinkage operator defines as follow

$$\mathcal{S}_{\theta_1',\theta_2',\rho}(x) = \begin{cases} x + \theta_1' + \theta_2', & x < -\theta_1' - \theta_2' \\ 0, & -\theta_1' - \theta_2' \leq x \leq \theta_1' - \theta_2' \\ x - \theta_1' + \theta_2', & \theta_1' - \theta_2' < x < \theta_1' - \theta_2' + \rho \\ \rho, & \theta_1' - \theta_2' + \rho \leq x \leq \theta_1' + \theta_2' + \rho \\ x - \theta_1' - \theta_2' & x > \theta_1' + \theta_2' + \rho \end{cases}$$

R EFERENCES

[1] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," in *Proc. IEEE.*, vol. 93, no. 1, pp. 71-83, 2005.

[2] E. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406-5425, Apr. 2006.

[3] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489-509, Feb. 2006.

[4] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289-1306, Apr. 2006.

[5] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "An architecture for compressive imaging," in *Proc. IEEE Int. Conf. Image Process.*, 2006, pp. 1273-1276.

[6] M. F. Durate, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G Baraniuk, "Single-pixel camera via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 83-91, Mar. 2008.

[7] M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "Compressive imaging for video representation and coding," in *Proc. PCS*, Apr. 2006, pp. 711-716.

[8] I. Drori, "Compressed video sensing," in *BMVA Symp. 3D Video-Anal., Display Applicat.*, 2008.

[9] J. Y. Park and M. B. Wakin, A multiscale framework for compressive sensing of video, in *Proc. PCS*, May. 2009, pp. 197-200.

[10] D. Reddy, A. Veeraraghavan, and R. Chellappa, "P2C2: programmable pixel compressive camera for high speed imaging," in *Proc. IEEE CVPR*, June 2011, pp. 329-336.

[11] M. Cossalter, G. Valenzise, M. Tagliasacchi, and S. Tubaro, "Joint compressive video coding and analysis," *IEEE Tran. Multimedia*, vol. 12, no. 3, pp. 168-183, Apr. 2010.

[12] V. Stanković, L. Stanković, and S. Cheng, "Compressive video sampling," in *Proc.16th Eur. Signal Process. Conf. (Eusipco-2008)*, Aug. 2008.

[13] J. Prades-Nobet, Y. MA, and T. Huang, "Distributed video coding using compressive sampling," in *Proc. PCS*, May 2009, pp. 165-168.

[14] T. T. Do, Y. Chen, D. T. Nguyen, N. Nguyen, L. Gan, and T. D. Tran, "Distributed compressed video sensing," in *Proc. IEEE ICIP*, Nov. 2009, pp. 1393-1396.

[15] H. W. Chen, L. W. Kang, and C. S. Lu, "Dynamic measurement rate allocation for distributed compressive video sensing," in *Proc. SPIE VCIP*, Jul. 2010, pp. 77401-77410.

[16] R.Marcia and R.Willet, "Compressive coded aperture video reconstruction," in *Proc.16th Eur. Signal Process. Conf. (Eusipco-2008)*, Aug. 2008.

[17] J. Zheng and E. Jacobs, "Video compressive sensing using spatial domain sparsity," *Optic. Eng.*, vol. 48, no. 8, pp. 1-10, Aug. 2009.

[18] C. Chen, E. W. Tramel, and J. E. Fowler, "Compressed sensing recovery of images and video using multihypothesis predictions," in *Proc. 45th Asilomar Conf. Signals Sys. Comput.*, Nov. 2011, pp. 1193-1198.

[19] H. W. Chen, L. W. Kang, and C. S. Lu, "Dictionary Learning-based distributed compressive sensing," in *Proc. PCS.*, Dec. 2010, pp. 210-213.

[20] H. Liu, B. Song, H. Qin, and Z. Qiu, "A dictionary generation scheme for block-based compressed video sensing," in *Proc. IEEE Int. Conf. Signal Process. commun. comput.*, Sept. 2011, pp. 670-674.

[21] ___, "Dictionary learning based reconstruction for distributed compressed video sensing," *J. Vis. Commun. Image R.*, vol. 24, no. 8, pp. 1232-1242, Nov. 2013.

[22] J. Ma, G. Plonka, and M. Y. Hussaini, "Compressive video sampling with approximate message passing decoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 9, pp. 1354-1364, Sept. 2012.

[23] Y. Liu, M. Li, and D. A. Pados, "Motion-aware decoding of compressed-sensing video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 3, pp. 438-444, Mar. 2013.

[24] T. Goldstein and S. Osher, "The split Bregman method for ℓ_1 regularized problems," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 323-343, Apr. 2009.

[25] J. F. Cai, S. Osher, and Z. W. Shen, "Split Bregman methods and frame based image restoration," *Multiscale Model. Simul.*, vol. 8, no. 2, pp. 5057-5071, Dec. 2009.

[26] J. Zhang, C. Zhao, D. Zhao, and W. Gao, "Image compressive sensing recovery using adaptively learned sparsifying basis via ℓ_0 minimization," *Signal Process.*, vol. 103, pp. 114-126, Oct. 2014.

[27] A. Beck and M. Tebulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183-202, Jan. 2009

[28] W. Yin, S. Osher, D. Goldfarm, and J. Darbon, "Bregman iterative algorithms for ℓ_1 minimization with applications to compressed sensing," *SIAM J. Imag. Sci.*, vol. 1, no. 1, pp. 143-168, Mar. 2008.

[29] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Learning overcomplete dictionaries based on atom-by-atom updating," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 883-891, Feb. 2014.

[30] K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *Proc. IEEE ICASSP*, Mar. 1999, vol. 5, pp. 2443-2446.

[31] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311-4322, Nov. 2006.

[32] L. N. Smith and M. Elad, "Improving dictionary learning: Multiple dictionary updates and coefficient reuse," *IEEE Signal Process. Lett.*, vol. 20, no. 1, pp. 79-82, Jan. 2013.

[33] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Trans. Image process.*, vol. 23, no. 8, pp. 3336-3351, Aug. 2014.

[34] J. A. Tropp and A. Gilbert, "Signal recovery from random measurement via orthogonal matching pursuit," *IEEE Trans. Inform. Theory*, vol. 53, no. 12, pp. 4655-4666, Dec. 2007.

[35] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33-61, Aug. 1998.

[36] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B.*, vol. 58, no. 1, pp. 267-288, Oct. 1996.

[37] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," *Adv. Neural Inform. Process. Syst.*, pp. 801-808, 2007.

[38] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure. Appl. Math.*, vol. 57, no. 11, pp. 1413-1457, Nov. 2004.

[39] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proc. IEEE CVPR*, 2011, pp. 457-464.

[40] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736-3745, Dec. 2006.

[41] Z. Wong, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600-612, Apr. 2004.

[42] N. G. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *J. Appl. comput. Harmon. Anl.*, vol. 10, no. 3, pp. 234-253, May 2001.

This paper was presented in part in 7th International Symposium on Telecommunications, Tehran, Iran, Sep. 2014.

N. Eslahi, A. Aghagolzadeh, and S. M. H. Andargoli, "Recovery of compressive video sensing via dictionary learning and forward prediction," in *Proc. 7th Int. Symp. Telecommun. (IST)*, Sep. 2014, pp. 833-838.